Object Oriented Programming

SPRING 2021 Solutions

1.A

i. Declare m as a static variable.
ii. Add the keyword final to method1() only inside class B.
iii. Add the keyword final to the declaration of class C.

```
class A {
	static int m = 10;
	void method1(int t) {
		System.out.println(t);
	}
	static void method2() {
		System.out.println(m);
	}
}

class B extends A {
	final void method1(int t) {
		System.out.println(t);
	}
}

final class C extends A {
	void method1(int t) {
		System.out.println(t);
	}
}
```

1.B

```
interface Vehicle {
	void fuelType();
}

public class Spring {
	public static void main(String[] args) {
		Vehicle cng;
		Vehicle airplane;

		cng = new Vehicle() {
			@Override
			public void fuelType() {
				System.out.println("Uses LPG");
			}
		};

		airplane = new Vehicle() {
			@Override
			public void fuelType() {
				System.out.println("Uses Jet Fuel");
```

```
                }
            };

            cng.fuelType();
            airplane.fuelType();
      }
}
```

2

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;

public class Main {
      public static void main(String[] args) {
            final String inputFile = "\\path";
            final String evenFile = "\\path";
            final String oddFile = "\\path";

            ArrayList<String> even = new ArrayList<>();
            ArrayList<String> odd = new ArrayList<>();

            try(FileReader fr = new FileReader(inputFile)) {
                  try(BufferedReader br = new BufferedReader(fr)) {
                        String buff;
                        long value = 0;
                        while(br.ready()) {
                              buff = br.readLine();
                              value = Long.parseLong(buff);
                              if(value % 2 == 0) {
                                    even.add(buff);
                              }
                              else {
                                    odd.add(buff);
                              }
                        }
                        br.close();
                  }
                  fr.close();
            }
            catch(NumberFormatException | IOException e) {
                  e.printStackTrace();
            }

            File ef = new File(evenFile);
            File of = new File(oddFile);

            try(PrintWriter pew = new PrintWriter(ef)) {
                  try(PrintWriter pow = new PrintWriter(of)) {
                        for(String s: odd) {
                              pow.println(s);
                        }
                        pow.close();
                  }

                  for(String s: even) {
                        pew.println(s);
                  }
```

```
                pew.close();
            }
            catch(IOException e) {
                e.printStackTrace();
            }

            odd.clear();
            even.clear();
        }
}
```

3

```
class InvalidTxnException extends Exception {
        public InvalidTxnException(String s) {
            super(s);
        }
}

class CreditCard {
        private double credit_limit;
        private double credit_current;
        public CreditCard(double limit) throws InvalidTxnException {
            // check and throw InvalidTxnException
            if(limit < 0) throw new InvalidTxnException(limit + " is not a valid amount
for the requested transaction.");
            credit_limit = limit;
            credit_current = 0;
        }

        public void withdraw(double amount) throws InvalidTxnException {
            // check and throw InvalidTxnException
            if(credit_limit - (credit_current + amount) < 0) {
                throw new InvalidTxnException(amount + " cannot be withdrawn with
current credit of " + credit_current + " for your limit of " + credit_limit);
            }
            credit_current += amount;
        }
}

public class Main {
        public static void main(String args[]) {
            // handle the proper exception here with try-catch
            try {
                CreditCard c1 = new CreditCard(-5000);
                CreditCard c2 = new CreditCard(10000);
                c2.withdraw( 7080);
                c2.withdraw(4000);
            }
            catch (InvalidTxnException e) {
                System.out.print(e.getMessage());
            }
        }
}
```

4

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        JFrame frame = new JFrame("My App");

        drawComponents(frame);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 300);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    public static void drawComponents(JFrame window) {
        JPanel panel = new JPanel(new GridLayout(0, 2));
        window.setContentPane(panel);

        ArrayList<JComponent> buttons = new ArrayList<>();
        buttons.add(new JButton("1"));
        buttons.add(new JButton("2"));
        buttons.add(new JButton("3"));
        buttons.add(new JButton("4"));

        panel.add(buttons.get(0));
        panel.add(buttons.get(1));
        panel.add(buttons.get(2));
        panel.add(buttons.get(3));

        ((JButton)buttons.get(2)).addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                ((JButton)buttons.get(0)).setText("1");
                ((JButton)buttons.get(1)).setText("3");
                ((JButton)buttons.get(2)).setText("2");
                ((JButton)buttons.get(3)).setText("4");
            }
        });
    }
}
```

5

```
import java.util.ArrayList;
import java.util.Comparator;

class Player {
      int jersey;
      String name, type;
      public Player(int jersey, String name, String type) {
            this.jersey = jersey;
            this.name = name;
            this.type = type;
      }
}

public class comparator_main {
      public static void main(String[] args) {
            ArrayList<Player> list = new ArrayList<>();

            list.add(new Player(55, "Karim", "Bangladesh"));
```

```
            list.add(new Player(14, "Ponting", "Australia"));

            list.sort(new Comparator<Player>() {
                  public int compare(Player p1, Player p2) {
                        if(p1.jersey > p2.jersey) {
                              return 1;
                        }
                        else if(p2.jersey > p1.jersey) {
                              return -1;
                        }
                        else {
                              return 0;
                        }
                  }
            });
      }
}
```

6